A Riemannian gradient flow perspective on learning deep linear neural networks

Ulrich Terstiege
Chair for Mathematics of Information Processing, RWTH Aachen University,
Pontdriesch 10, 52062 Aachen, Germany
terstiege@mathc.rwth-aachen.de

Holger Rauhut
Chair for Mathematics of Information Processing, RWTH Aachen University,
Pontdriesch 10, 52062 Aachen, Germany
rauhut@mathc.rwth-aachen.de

Bubacarr Bah
African Institute for Mathematical Sciences (AIMS) South Africa
Stellenbosch University, 6 Melrose Road, Muizenberg, Cape Town 7945, South Africa
bubacarr@aims.ac.za

Michael Westdickenberg
Institute for Mathematics, RWTH Aachen University,
Templergraben 55, 52062 Aachen, Germany
mwest@instmath.rwth-aachen.de

Abstract

We study the convergence of gradient flows related to learning deep linear neural networks from data. In this case, the composition of the network layers amounts to simply multiplying the weight matrices of all layers together, resulting in an overparameterized problem. The gradient flow with respect to these factors can be re-interpreted as a Riemannian gradient flow on the manifold of rank-r matrices endowed with a suitable Riemannian metric. We show that the flow always converges to a critical point of the underlying functional. Moreover, we establish that, for almost all initializations, the flow converges to a global minimum on the manifold of rank k matrices for some $k \leq r$.

1 Introduction

Training a deep neural network amounts to solving an empirical risk minimization problem. One commonly uses gradient descent methods for this task. While this is the work horse of current deep learning technology and works very well in practice, the theoretical understanding of this approach is lacking to a large extent, mainly due to the nonconvexity of the underlying optimization problem. In this work, see [1] for all technical details, we study convergence to the (global) minimizers of the corresponding objective functional. In the context of general *nonlinear* networks this problem seems to be very involved and therefore we focus on the simpler case of *linear* networks, building on previous work in e.g. [5, 3, 2, 4, 7]. While the class of linear neural networks may not be rich enough for many machine learning tasks, it is nevertheless instructive and still a non-trivial task to understand the convergence properties of gradient descent algorithms.

NeurIPS 2020 workshop on Differential Geometry meets Deep Learning

Our contributions can be summarized as follows:

- We show that in the balanced case (see definition 2) the evolution of the product of all network layer matrices can be re-interpreted as a Riemannian gradient flow on the manifold of matrices of some fixed rank. This reveals an unexpected connection of deep learning to Riemanian geometry.
- We show that the flow always converges to a critical point of the loss functional L^N, see (2). This results applies under significantly more general assumptions than results in [4].
- We show that the flow converges to the global optimum of the loss functional L^1, see (4), restricted to the manifold of rank k matrices for almost all initializations (Theorem 5), where the rank may be anything between 0 and r (the smallest of the involved matrix dimensions). In the case of two layers, we show in the same theorem that for almost all initial conditions, the flow converges to a global optimum of L^2, see (2). For the proof, we extend an abstract result in [6] from gradient descent to gradient flows establishing that strict saddle points of the functional are avoided for almost all initializations.

2 Gradient flows for learning linear networks

We consider the regression problem where we are given m data points $x_1, \dots, x_m \in \mathbb{R}^{d_x}$ and label points $y_1, \dots, y_m \in \mathbb{R}^{d_y}$ and would like to find a map f such that $f(x_j) \approx y_j$. In deep learning, candidate maps are given by deep neural networks of the form

$$f(x) = f_{W_1,\dots,W_N,b_1,\dots,b_N}(x) = g_N \circ g_{N-1} \circ \cdots \circ g_1(x),$$

where each layer is of the form $g_j(z) = \sigma(W_j z + b_j)$ with matrices W_j and vectors b_j and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ that acts componentwise. Here, we concentrate on the simplified setting of linear networks of the form

$$f(x) = W_N \cdot W_{N-1} \cdots W_1 x, \quad \text{for } N \geq 2,$$

where $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ for $d_0 = d_x$, $d_N = d_y$ and $d_1, \dots, d_{N-1} \in \mathbb{N}$. Then $f(x) = Wx$ with the factorization

$$W = W_N \cdots W_1, \tag{1}$$

which can be viewed as an overparameterization of the matrix W. Note that the factorization imposes a rank constraint as the rank of W is at most $r = \min\{d_0, d_1, \dots, d_N\}$.

In order to fit the linear network to the data, we consider minimizing the squared loss, i.e., the functional

$$L^N(W_1, \dots, W_N) = \frac{1}{2} \sum_{j=1}^{m} \|y_j - W_N \cdots W_1 x_j\|_2^2 = \frac{1}{2} \|Y - W_N \cdots W_1 X\|_F^2 \tag{2}$$

where $X \in \mathbb{R}^{d_x \times m}$ is the matrix with columns $x_1, \dots, x_m$ and $Y \in \mathbb{R}^{d_y \times m}$ the matrix with columns $y_1, \dots, y_m$. Here $\| \cdot \|_F$ denotes the Frobenius norm induced by the inner product $\langle A, B \rangle_F := \operatorname{tr}(AB^T)$.

Empirical risk minimization is the optimization problem

$$\min_{W_1,\dots,W_N} L^N(W_1, \dots, W_N), \quad \text{where } W_j \in \mathbb{R}^{d_j \times d_{j-1}},\ j = 1, \dots, N. \tag{3}$$

For $W \in \mathbb{R}^{d_y \times d_x}$, we further introduce the functional

$$L^1(W) := \frac{1}{2} \|Y - WX\|_F^2. \tag{4}$$

Since the rank of $W = W_N \cdots W_1$ is at most $r = \min\{d_0, d_1, \dots, d_N\}$, minimization of L^N is closely related to the minimization of L^1 restricted to the set of matrices of rank at most r, but the optimization of L^N does not require to formulate this constraint explicitly. However, L^N is not jointly convex in $W_1, \dots, W_N$ so that understanding the behavior of corresponding optimization algorithms is not trivial.

The gradient of L^1 is given as

$$\nabla_W L^1(W) = WXX^T - YX^T.$$

For given initial values $W_j(0),\ j \in \{1,\dots,N\}$, we consider the system of gradient flows

$$\dot{W}_j = -\nabla_{W_j} L^N(W_1,\dots,W_N),\ j = 1,\dots,N, \tag{5}$$

i.e., these flows run in a synchronous way. In this work we investigate when this system converges to an optimal solution, i.e., one that is minimizing our optimization problem (3). For $W = W_N \cdots W_1$ we also want to understand the behavior of $W(t)$ as t tends to infinity. Clearly, the gradient flow is a continuous version of gradient descent algorithms used in practice and has the advantage that its analysis does not require discussing step sizes.

Our first main result [1, Theorem 5] establishes convergence of the gradient flow.

Theorem 1. *Assume XX^T has full rank. Then the flows $W_i(t)$ defined by (5) are defined and bounded for all $t \geq 0$ and $(W_1(t),\dots,W_N(t))$ converges to a critical point of L^N as $t \to \infty$.*

3 Riemannian gradient flows

It turns out that the product matrix $W(t) = W_N(t)\cdots W_1(t)$ formed with the individual matrices of the gradient flow satisfies an interesting relation that can be connected to a Riemannian gradient flow in the case that the $W_1,\dots,W_N$ are balanced as defined next, cf. [2].

Definition 2. We say that $W_1,\dots,W_N$ with $W_j \in \mathbb{R}^{d_j \times d_{j-1}},\ j = 1,\dots,N$, are *balanced* if

$$W_{j+1}^T W_{j+1} = W_j W_j^T \text{ for } j = 1,\dots,N-1.$$

We say that the flow (5) has balanced initial conditions if $W_1(0),\dots,W_N(0)$ are balanced.

If the $W_j(0)$ are balanced and the $W_j(t)$ satisfy the flow (5), then one easily checks that $W_1(t),\dots,W_N(t)$ are balanced for any $t \geq 0$, cf. [3]. It was first established in [3] that then the flow for the product $W(t) = W_N \cdots W_1(t)$ is given by

$$\frac{dW(t)}{dt} = -\mathcal{A}_{W(t),N}\Big(\nabla_W L^1\big(W(t)\big)\Big), \tag{6}$$

where for $W, Z \in \mathbb{R}^{d_y \times d_x}$ and $N \geq 2$ the operator $\mathcal{A}_{W,N}$ is defined as

$$\mathcal{A}_{W,N}(Z) = \sum_{j=1}^{N} (WW^T)^{\frac{N-j}{N}} \cdot Z \cdot (W^TW)^{\frac{j-1}{N}}. \tag{7}$$

The product matrix $W(t)$ clearly is of rank at most $r := \min\{d_i : i = 0,\dots,N\}$. Therefore, we introduce the (differentiable) manifold $\mathcal{M}_k$ of real $d_y \times d_x$ matrices of rank $k \leq \min\{d_x, d_y\}$ and denote by $T_W(\mathcal{M}_k)$ the tangential space of $\mathcal{M}_k$ at the point $W \in \mathcal{M}_k$. The operator $\mathcal{A}_{W,N}$ induces a Riemannian metric on $\mathcal{M}_k$ as outlined in the next main result [1].

Theorem 3. *The map $\mathcal{A}_{W,N}$ is self-adjoint with image $T_W(\mathcal{M}_k)$ and the restriction of $\mathcal{A}_{W,N}$ to arguments $Z \in T_W(\mathcal{M}_k)$ defines a self-adjoint and positive definite map*

$$\bar{\mathcal{A}}_{W,N} \colon T_W(\mathcal{M}_k) \to T_W(\mathcal{M}_k).$$

In particular, $\bar{\mathcal{A}}_{W,N}$ is invertible and the inverse $\bar{\mathcal{A}}_{W,N}^{-1}$ is self-adjoint and positive definite as well. Hence,

$$g_W(Z_1, Z_2) := \langle \bar{\mathcal{A}}_{W,N}^{-1}(Z_1), Z_2\rangle_F, \quad W \in \mathcal{M}_k, Z_1, Z_2 \in T_W(\mathcal{M}_k), \tag{8}$$

is well-defined. It is a Riemannian metric of class C^1 on $\mathcal{M}_k$, which is explicitly given by the expression

$$g_W(Z_1, Z_2) = \frac{\sin(\pi/N)}{\pi} \int_0^\infty \operatorname{tr}\left((tI_{d_y} + WW^T)^{-1} Z_1 (tI_{d_x} + W^TW)^{-1} Z_2^T\right) t^{1/N} dt.$$

It is not clear at the moment whether the metric is also C^2.

Given a Riemannian metric g on $\mathcal{M}_k$ we can introduce the Riemannian gradient ∇^g of a differentiable function f on $\mathcal{M}_k$ at $W \in \mathcal{M}_k$ via

$$g_W(\nabla^g f(W), Z) = df(W)[Z] \quad \text{for all } Z \in T_W(\mathcal{M}_k),$$

where $df(W) : T_W(\mathcal{M}_k) \to \mathbb{R}$ denotes the standard differential of f at W. In view of (8) we have

$$\nabla^g f(W) = \mathcal{A}_{W,N}(\nabla f(W)),$$

where ∇f is the standard (Euclidean) gradient of f (which satisfies $\langle \nabla f(W), Z\rangle = df(W)[Z]$ with the Frobenius scalar product $\langle \cdot, \cdot \rangle$). With this we can express $W(t)$ as the solution of a Riemannian gradient flow equation as stated next.

Theorem 4. *1. Assume that XX^T has full rank and suppose that $W_1(t), \dots, W_N(t)$ are solutions of the gradient flow (5) of L^N with balanced initial values $W_j(0)$. For $t \geq 0$ define the product $W(t) := W_N(t) \cdots W_1(t)$ and let $k \leq \min\{d_0, \dots, d_N\}$ be the rank of $W(0)$. Then $W(t)$ is contained in $\mathcal{M}_k$ for all $t \geq 0$ and uniquely solves the gradient flow equation*

$$\dot{W} = -\nabla^g L^1(W) \quad \text{on } \mathcal{M}_k \quad \text{for all } t \in [0, \infty). \tag{9}$$

2. Assume that XX^T has full rank and let $N \geq 2$. Then for any initialization $W(0) \in \mathbb{R}^{d_y \times d_x}$, denoting by k the rank of $W(0)$, there is a uniquely defined flow $W(t)$ on $\mathcal{M}_k$ for $t \in [0, \infty)$ which satisfies (9).

4 Convergence to global minimizers

While convergence to critical points of the gradient flow has been established in Theorem 1, it is not clear at this point whether we can expect convergence to a global minimizer of L^N. The next main result [1, Theorem 39] provides some insights to this question.

Theorem 5. *Assume that XX^T has full rank, let $q = \operatorname{rank}(YX^T(XX^T)^{-\frac{1}{2}})$, $r = \min\{d_0, \dots, d_N\}$ and let $\bar{r} := \min\{q, r\}$.*

(a) For almost all initial values $W_1(0), \dots, W_N(0)$, the flow (5) converges to a critical point $(W_1, \dots, W_N)$ of L^N such that $W := W_N \cdots W_1$ is a global minimizer of L^1 on the manifold $\mathcal{M}_k$ of matrices in $\mathbb{R}^{d_N \times d_0}$ of rank $k := \operatorname{rank}(W)$, where k lies between 0 and $\bar{r}$ and depends on the initialization.

(b) For $N = 2$, for almost all initial values $W_1(0)$ and $W_2(0)$, the flow (5) converges to a global minimizer of L^N on $\mathbb{R}^{d_0 \times d_1} \times \mathbb{R}^{d_1 \times d_2}$.

Here, *"for almost all"* means for all up to a set of measure zero. We conjecture that also for $N \geq 3$ the flow converges to a global minimizer of L^N for almost all initializations.

The next result [1, Theorem 40] directly analyzes the corresponding Riemannian gradient flow (9) on $\mathcal{M}_k$. This corresponds to balanced initial conditions for the flow $(W_1(t), \dots, W_N(t))$, but since the balanced tuples of matrices $(W_1, \dots, W_N)$ form a set of zero themselves, the result below cannot be deduced directly from the previous theorem.

Theorem 6. *Assume that XX^T has full rank and let $N \geq 2$. Then for almost all initializations $W(0) \in \mathbb{R}^{d_y \times d_x}$ on $\mathcal{M}_k$, the flow $W(t)$ on $\mathcal{M}_k$ solving (9) (cf. Theorem 4) converges to a global minimum of L^1 restricted to $\mathcal{M}_k$ or to a critical point on some $\mathcal{M}_\ell$, where $\ell < k$. Note that for $k > \operatorname{rank}((YX^T(XX^T)^{-\frac{1}{2}}))$ there is no global minimum of L^1 on $\mathcal{M}_k$ so that then the second option applies. Here again,* "for almost all $W(0)$" *means for all $W(0)$ up to a set of measure zero.*

Both results are established by characterizing the strict saddle points of the functionals L^N, and L^1, see [1, Section 6.4] and [5, 7], which are defined as critical points, where the (Riemannian) Hessian has a negative eigenvalue, and by showing that Riemannian gradient flows avoid strict saddle points for almost all initializations, see [1, Theorem 28]. The latter result extends a corresponding result in [6] from gradient descent iterations to the case of gradient flows.

APPENDIX: Experimental results

We numerically study the convergence of gradient flows in the linear supervised learning setting as a proof of concept of the convergence results presented above in both the general supervised learning case and the special case of autoencoders.

A General supervised learning case

We start with experiments to test the results in the general supervised learning setting to support theoretical results in Theorems 5 and 6. We show results for $N = 2, 5, 10, 20$, and two sets of values for d_x and r (rank of $W(t)$ and $\widetilde{W}$, the true parameters). The data matrix X is generated as in the autoencoder case and $Y = \widetilde{W}X$, where $\widetilde{W} = \widetilde{W}_N \cdots \widetilde{W}_1$, with $\widetilde{W}_j \in \mathbb{R}^{d_j \times d_{j-1}}$ for $j = 1, \ldots, N$ with $d_N = d_0 = d_x = d$ and $d_1 = r$ is the rank of $\widetilde{W}$. The entries of $\widetilde{W}_j$ are randomly generated independently from a Gaussian distribution with standard deviation $\sigma = 1/\sqrt{d_j}$. The dimensions $d_j \times d_{j-1}$ of the W_j for $j = 1, \ldots, N$, are again selected respectively in an integer grid, i.e., $d_j = [r + (d_x - r)(j-1)/(N-1)]$, where r is arbitrarily fixed. The initial conditions are generated as was done in the autoencoder case. We investigate the convergence rates for the

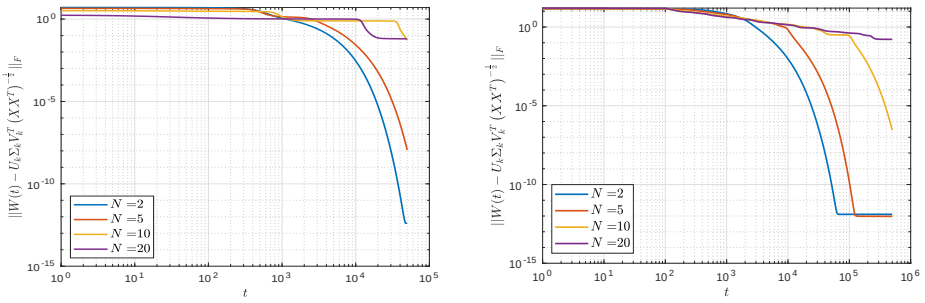

Figure 1: Convergence rates of solutions of the gradient flow of the general supervised learning problem depicted by convergence to critical points $U_k\Sigma_k V_k (XX^T)^{-\frac{1}{2}}$ with balanced initial conditions for *left panel:* $d_x = 20$, $r = 2$; *right panel:* $d_x = 200$, $r = 20$.

balanced and non-balanced initial conditions of the gradient flows. The results of the experiments are plotted in Figures 1 and 2. In these plots k is the rank of $Q := YX^T(XX^T)^{-\frac{1}{2}} \in \mathbb{R}^{d_y \times d_x}$, and $Q = U_k\Sigma_k V_k$ is the (reduced) singular value decomposition of Q, i.e., $U_k \in \mathbb{R}^{d_x \times k}$ and $V_k \in \mathbb{R}^{d_y \times k}$ have orthonormal columns and $\Sigma_k \in \mathbb{R}^{k \times k}$ is a diagonal matrix containing the non-zero singular values of Q.

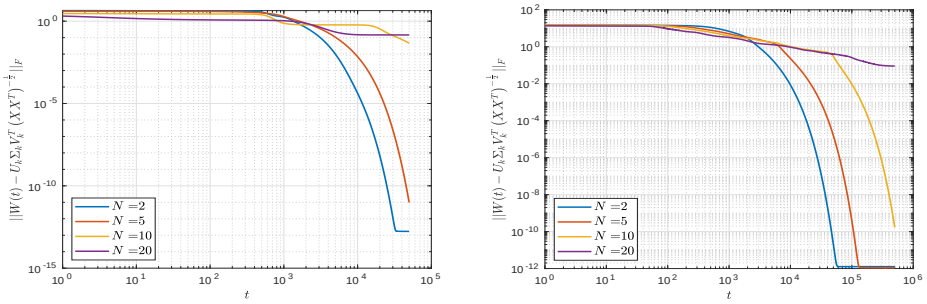

Figure 2: Convergence rates of solutions of the gradient flow of the general supervised learning problem depicted by convergence to critical points $U_k\Sigma_k V_k (XX^T)^{-\frac{1}{2}}$ with non-balanced initial conditions for *left panel:* $d_x = 20$, $r = 2$; *right panel:* $d_x = 200$, $r = 20$.

With balanced initial conditions the plots of Figure 1 show convergence rates of the flow to critical points $U_k\Sigma_k V_k (XX^T)^{-\frac{1}{2}}$ (as explicitly stated in [1, Proposition 32]). Similaryly, with non-balanced initial conditions the plots of Figure 2 show convergence rates to these critical points. These results show rapid convergence of the flow and the dependence of the convergence rate on N, r and d_x with

either balanced or non-balanced initial conditions. Note that to critical points $U_k \Sigma_k V_k (XX^T)^{-\frac{1}{2}}$ (as explicitly stated in [1, Proposition 32]) are the same as the true parameters $\widetilde{W}$. This can be seen by comparing the left panel plot of Figure 1 to the left panel plot of Figure 3 and the left panel plot of Figure 2 to the right panel plot of Figure 3.

Convergence is slower for larger N, and it seems not to depend on the initial conditions, balanced or non-balanced, see the plots of Figures 1 and 2. Equivalently, this can be seen from the error of the supervised learning loss shown in the plots of Figure 4 for balanced initial conditions. There is much stronger dependence on N in this setting than in the autoencoder setting.

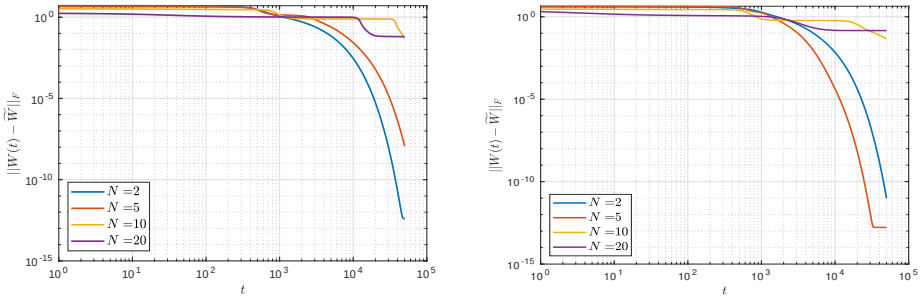

Figure 3: Convergence to the true parameters $\widetilde{W}$ for ($d_x = 20$, $r = 2$) with *left panel:* balanced initial conditions; *right panel:* non-balanced initial conditions.

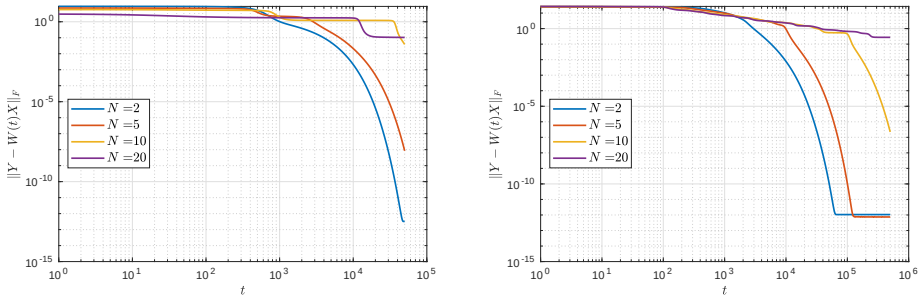

Figure 4: General supervised learning errors with balanced initial conditions for dimensions *left panel:* $d_x = 20$, $r = 2$; *right panel:* $d_x = 200$, $r = 20$.

B Autoencoder case

The results of this paper (precisely Theorems 5 and 6) also hold for the autoencoder setting, where $Y = X \in \mathbb{R}^{d_x \times m}$ in (3). We thus study here the gradient flow (5) in the autoencoder setting, for different dimensions of X (i.e., d_x and m) and different values of the number N of layers, where we typically use $N \in \{2, 5, 10, 20\}$. A Runge-Kutta method (RK4) is used to solve the gradient flow differential equation with appropriate step sizes $t_n = t_0 + nh$ for large n and $h \in (0, 1)$. The experiments fall into two categories based on initial conditions of the gradient flow: a) *balanced* – where the balanced conditions are satisfied; and b) *non-balanced* – where the balanced conditions are not satisfied.

The results in summary, considering $W = W_N \cdots W_1$ as the limiting solution of the gradient flow, that is $W = \lim_{t\to\infty} W(t)$, where $W(t) = W_N(t) \cdots W_1(t)$: we show that with balanced initial conditions, the solutions of the gradient flow converges to $U_r U_r^T$, where the columns of U_r are the r eigenvectors corresponding to the r largest eigenvalues of XX^T. The convergence rates decrease with an increase in either d or N or both. We see similar results for the non-balanced case.

B.1 Balanced initial conditions

In this section and Section B.2 the data matrix $X \in \mathbb{R}^{d_x \times m}$ is generated with columns drawn i.i.d. from a Gaussian distribution, i.e., $x_i \sim \mathcal{N}(0, \sigma^2 I_{d_x})$, where $\sigma = 1/\sqrt{d_x}$. Random realization of

X with sizes $d_x = d$ and $m = 3d$ are varied to investigate different dimensions of the input data, i.e., with $2N \leq d \leq 20N$. For each fixed d, the dimensions d_j of the $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ for $j = 1, \ldots, N$ are selected as follows: We set $d_1 = r = [d/2]$, where $[\cdot]$ rounds to the nearest integer, and put $d_j = [r + (d-r)(j-1)/(N-1)]$, $j = 2, \ldots, N$ (generating an integer "grid" of numbers between $d_1 = r$ and $d_N = d_x = d$).

In the first set of experiments, we consider a general case of the balanced initial conditions, precisely $W_{j+1}^T(0)W_{j+1}(0) = W_j(0)W_j^T(0)$, $j = 1, \ldots, N-1$. The dimensions of the W_j and their initializations are as follows. Recall, $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ for $j = 1, \ldots, N$ where $d_N = d_0 = d_x = d$ and $d_1 = r$ is the rank of $W = W_N \cdots W_1$. We randomly generate $d_j \times d_j$ orthogonal matrices V_j and then form $W_j(0) = V_j I_{d_j d_1} U_{j-1}^T$ for $j = 1, \ldots, N$, where $U_j \in \mathbb{R}^{d_j \times d_1}$ is composed of the d_1 columns of V_j, and I_{ab} is the (rectangular) $a \times b$ identity matrix. For all the values of N and the different ranks of W considered, Figure 5 shows that the limit of $W(t)$ as $t \to \infty$ is $U_r U_r^T$, where the columns of U_r are r eigenvectors of XX^T corresponding to the largest r eigenvalues of XX^T. Experiments were ran for $N = 2, 5, 10, 20$, but for the purpose of space we show results for $N = 2$ and $N = 20$. This agrees with the theoretical results in Theorems 5 and 6 for the autoencoder setting.

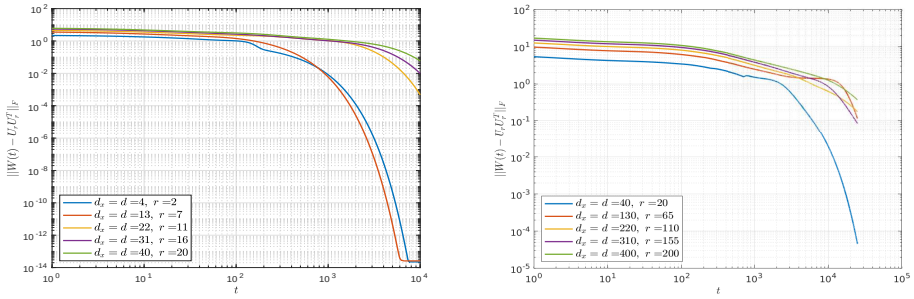

Figure 5: Convergence of solutions for the general balanced case. Error between $W(t)$ and $U_r U_r^T$ for different r and d values. *Left panel*: $N = 2$; *right panel*: $N = 20$.

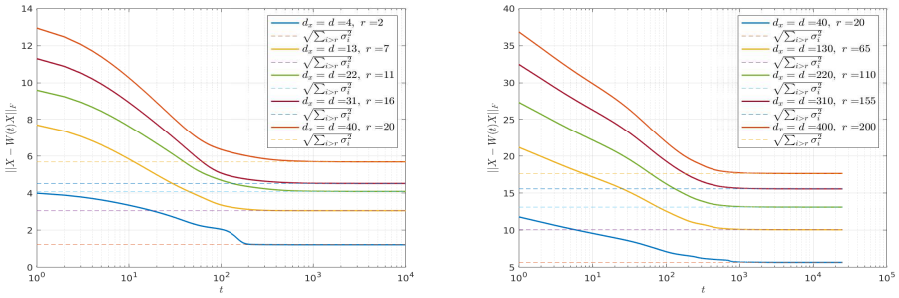

Figure 6: Convergence of solutions for the general balanced case. Errors between X and $W(t)X$ for different r and d values. *Left panel*: $N = 2$; *right panel*: $N = 20$.

In addition, when $W(t)$ converges to $U_r U_r^T$ then $\|X - W(t)X\|_F$ converges to $\sqrt{\sum_{i>r} \sigma_i^2}$. This is also tested and confirmed for $N = 2, 5, 10, 20$, but for the purpose of space we show results for $N = 2$ and $N = 20$ in Figure 6. This depicts convergence of the functional $L^1(W(t))$ to the optimal error, which is the square-root of the sum of the tail eigenvalues of XX^T of order greater than r.

B.2 Non-balanced initial conditions

For $W_j(0)$, $j = 1, \ldots, N$, we randomly generate Gaussian matrices. As in the balanced case we see that $W(t)$ converges to $U_r U_r^T$. This is also tested and confirmed for $N = 2, 5, 10, 20$, but for the purpose of saving space we show results for $N = 2$ and $N = 20$ in Figure 7.

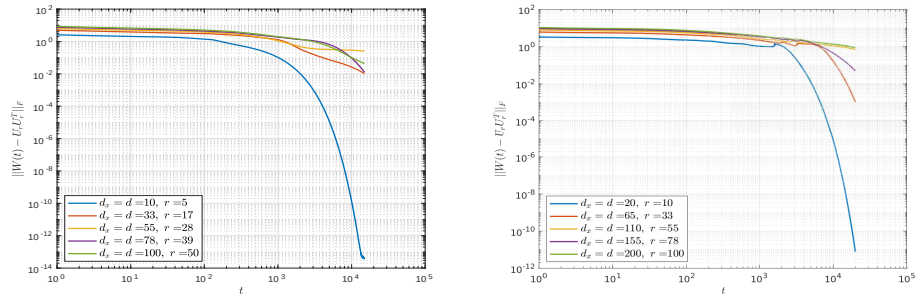

Figure 7: Non-balanced case convergence of solutions of the gradient flow. Errors between $W(t)$ and $U_r U_r^T$ for different r and d values for *left panel:* $N = 5$, *right panel:* $N = 10$.

B.3 Convergence rates

In order to better understand convergence rates we modify the previous experiments slightly. Here the data matrix $X \in \mathbb{R}^{d_x \times m}$ is generated with columns drawn i.i.d.from a Gaussian distribution, i.e., $x_i \sim \mathcal{N}(0, \sigma^2 I_{d_x})$, where $\sigma = 1/\sqrt{d_x}$. Random realization of X with two different values for d_x (as in above $m = 3d$) and different r, the rank of $W(t)$, are used. For each fixed d, the dimensions d_j of the $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ are selected using an arbitrarily chosen r and setting $d_j = [r + (d - r)(j - 1)/(N - 1)]$ for $j = 1, \ldots, N$. The value of r is stated in the caption of the figures. The experiments show very rapid convergence of the solutions but also the dependence of the convergence rate on N, d_x, and r. We investigate this for different values of N, d_x and r, in both the balanced and non-balanced cases. Convergence plots for the balanced initial conditions are shown in Figure 8, depicting smooth convergence. Similarly, we have convergence rates of the non-balanced case in Figure 9. These plots also show a slightly faster convergence for the balanced case than for the non-balanced case.

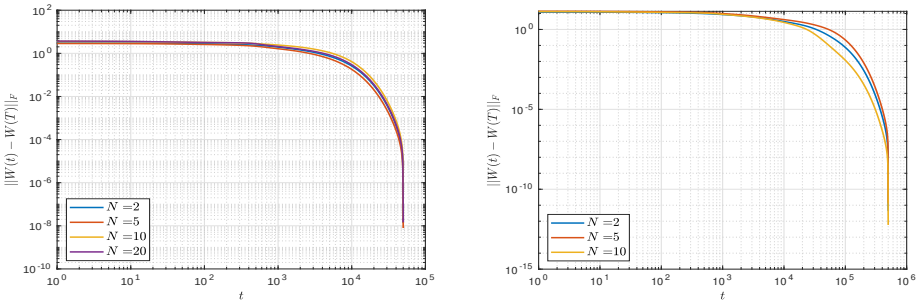

Figure 8: Convergence rates of solutions of the gradient flow in the autoencoder case with balanced initial conditions – errors between $W(t)$ and $W(T)$ for different N values, where T is the final time. Dimensions *Left panel*: $d_x = 20$, $r = 1$; *Right panel*: $d_x = 200$, $r = 10$.

Conclusion

In conclusion, in the autoencoder case we confirmed that the solutions of the gradient flow converges to $U_r U_r^T$, while in the general supervised learning case we confirmed convergence of the flow to critical points explicitly defined in [1, Proposition 32]. Such convergence occurs with either balanced or non-balanced initial conditions albeit a slight faster convergence in the balanced than in the non-balanced. Moreover, in both the autoencoder and the general supervised learning setting we see that as the size (N, d_x, r) of the problem instance increases the convergence rates decrease. In the autoencoder case we saw stronger dependence in d_x and r than in the general supervised learning case. On the other hand the dependence on N seems to be stronger in the general supervised learning case than in the autoencoder case.

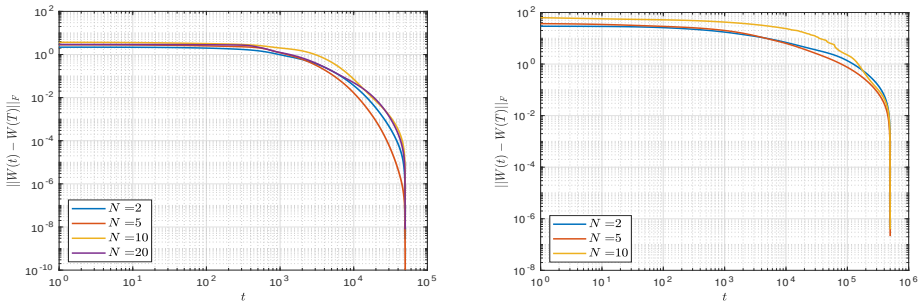

Figure 9: Convergence rates of solutions of the gradient flow in the autoencoder case with non-balanced initial conditions – errors between $W(t)$ and $W(T)$ for different N values, where T is the final time. Dimensions *Left panel*: $d_x = 20$, $r = 1$; *Right panel*: $d_x = 200$, $r = 10$.

Acknowledgment

B.B., H.R. and U.T. acknowledge funding through the DAAD project *Understanding stochastic gradient descent in deep learning* (project number 57417829). B.B. acknowledges funding by BMBF through the Alexander-von-Humboldt Foundation.

References

[1] B. Bah, H. Rauhut, U. Terstiege, M. Westdickenberg. *Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers* Information and Inference, to appear. arXiv:1910.05505

[2] S. Arora, N. Cohen, N. Golowich, and W. Hu. A convergence analysis of gradient descent for deep linear neural networks, ICLR, 2019. (arXiv:1810.02281).

[3] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *Preprint arXiv:1802.06509*, 2018.

[4] Y. Chitour, Z. Liao, and R. Couillet. A geometric approach of gradient descent algorithms in neural networks. *Preprint*, arXiv:1811.03568, 2018.

[5] K. Kawaguchi. Deep learning without poor local minima. *Advances in Neural Information Processing Systems 29*, pages 586–594, 2016.

[6] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht. First-order methods almost always avoid strict saddle points. *Math. Program.*, 176(1):311–337, 2019.

[7] M. Trager, K. Kohn, J. Bruna, *Pure and spurious critical points: a geometric study of linear networks.* Preprint arXiv:1910.01671, 2019.